UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/633,076	08/01/2003	Jennifer Melin	200308666-1	4237

22879 7590 08/19/2009
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
FORT COLLINS, CO 80528

EXAMINER	
DICKERSON, CHAD S	
ART UNIT	**PAPER NUMBER**
2625	
NOTIFICATION DATE	**DELIVERY MODE**
08/19/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
jessica.l.fusek@hp.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/633,076
Filing Date: August 01, 2003
Appellant(s): MELIN ET AL.

Philip S. Lyren
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 5/26/2009 appealing from the Office action mailed 11/26/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6426798	Yeung	7-2002
2003/0126219	Tanimoto	7-2003
2005/0179921	Brossman et al.	8-2005
2003/0048470	Garcia	3-2003
2003/0182367	Ohara	9-2003
6820067	Hammond et al.	11-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 112

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1-15 and 18 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Re claims 1, 11 and 18: The phrase in independent claims 1, 11 and 18 stating "*wherein the markup language code can enable an active user interface*" renders the claims indefinite. Since the Applicant has introduced two kinds of markup language to the claims (supported and unsupported), the Examiner would like to pose a question as to which of the two markup languages enable an active user interface? The Examiner would like more clarification regarding this claim feature. In light of examination, the Examiner will give the broadest reasonable interpretation of the claim features. The claim 2-10 and 12-15 are also rejected because of their dependency.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-4, 6, 9-11, 13, 15, 16 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yeung '798 (USP 6426798) in view of Brossman '921 (US Pub No 2005/0179921) and Tanimoto '219 (US Pub No 2003/0126219).

Re claim 1: Yeung '798 discloses a data structure for printer description file, comprising the steps of:

receiving a request for the printing device's configuration attributes at the printing device and the request is received from a requesting device (i.e. the request or query for the printing device's attributes occurs when a printer driver on the computer (40) accesses a printer-specific data structure on an external printer and compares this data structure to the universal printer data structure definition, which is stored on the requesting or querying computer. The printer-specific data structure or universal data structure, illustrated in figure 3, is a plurality of predetermined data elements used for storing various capabilities supported by one of a plurality of printers; see figs. 1-4 and 6; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24);

identifying markup language code embedded in the printing device associated with the configuration attributes supported by the printing device (i.e. in the system, the printing attributes are automatically mapped to an XML structure that arranges the printing attributes in a hierarchal order. When using the example of figure 6 to determine if a printer-specific data structure is valid, the attributes are compared to the attributes in the universal printer data structure definition. The comparison involves the attributes and the markup language associated with the attributes. The computing equipment accesses the EEPROM (132) of the printer to identify the universal printer description file (140) for configuration of the printer driver (14). The universal printer description file is considered as markup language code that is associated with the configuration attributes of the printer. Since this file is stored in the printer, this can also be considered as having the markup language that makes up the file to be embedded in the printer; see figs. 3-6; col. 5, line 23 – col. 6, line 17, col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24) and markup language code embedded in the printing device unsupported by the printing device (i.e. in the system, the UI constraints are identified regarding the printing device and these printer features that are not supported in the printer, but are identified by a recognizing these prevented features of XML. As seen in columns 7 and 8, there are features that may or may not be supported by the printing device. If the data element designates that the feature is not supported, this is an example of code embedded in the printing device as unsupported XML being identified; see col. 8, ln 1-62); and

transmitting the markup language code that is associated with the configuration attributes supported by the printing device, from the printing device to the requesting device (i.e. when the computer (40) used in the system accesses the printer-specific data structure through a communication line (106) to the printer (50), after it is discovered that the printer-specific data structure is valid, the data structure is sent or transmitted to the computer (40) for the printer driver to correctly communicate with the printer (50) using the printer-specific data structure. Also, during the initialization of the printer driver, the computer may access the memory of the printer to obtain the UPDF and the UPDF is transmitted to the computer; see figs. 1-3 and 6; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24) and excluding the markup language code that is unsupported by the printing device (i.e. in the system, the UI constraints are used to prevent the user from selecting capabilities or functions that are not supported by the printer. With the prevention of selecting these options, these options are excluded from the user in the system. The unsupported functions can be the TonerSaves function that may be designated as not supported by the printing device; see col. 8, ln 1-62).

However, Yeung ‘798 fails to specifically teach making a run-time determination in the printing device of the configuration attributes supported by the printing device.

However, this is well known in the art as evidenced by Brossman ‘921. Brossman ‘921 discloses making a run-time determination in the printing device of the configuration attributes supported by the printing device (i.e. the system of Yeung is similar to the system of Brossman in that a computer communicates with a printer regarding the printer's capabilities (same field of endeavor). In the use of the information handling system that can be incorporated in the printer, the capabilities of the printer are compared to the selected option by the user. The information handling system makes the determination of the attributes supported by the printer and performs the function of notifying a user the incapability of performing the function or takes the print job and performs the requested function. The function of the printer can happen at run-time since this function occurs when the printer is not in a time of designing the markup language associated with the printer capabilities. Also, the printer capabilities are represented by the XML format in an XML file; see figs. 1, 2 and 4; paragraphs [0015]-[0020], [0023] and [0027]-[0037]).

Therefore, in view of Brossman ‘921, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of making a run-time determination in the printing device of the configuration attributes supported by the printing device in order to see if the printer capabilities are available (as stated in Brossman ‘921 paragraph [0036]).

However, the combination of Yeung ‘798 and Brossman ‘921 fails to teach the feature of wherein the markup language code can enable an active user interface.

However, this is well known in the art as evidenced by Tanimoto ‘219. Tanimoto ‘219 discloses the feature of wherein the markup language code can enable an active user interface (i.e. Like the above references, a printer device communicates information about the printer to a computer (same field of endeavor). In the system, the printer is used to send device-setting form data in the HTML or XML form to the requesting device (2), or the client terminal. The client terminal uses the device-setting form data to display a window shown in figure 5 to the user. This is an example of the device-setting form data in HTML or XML being compiled and used to create an active user interface for the user to interact with; see figs. 1-5; paragraphs [0032]-[0036]).

Therefore, in view of Tanimoto ‘219, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein the markup language code can enable an active user interface incorporated in the device of Yeung ‘798, in combination with the features of Brossman ‘921, in order to have the client terminal decode the device-setting form data and show a general browser display to the user (as stated in Tanimoto ‘219 paragraph [0034]).

Re claim 2: The teachings of Yeung '798 in view of Brossman ‘921 and Tanimoto ‘219 are disclosed above.

Yeung '798 discloses a method, wherein the markup language code that is unsupported by the printing device is excluded by disabling links to the markup language code that is unsupported by the printing device (i.e. in the system, the user is prevented from selecting a capability or function that the printer cannot support. In this example, the system disables the option to access a certain option that is not supported. For example, if a face direction of a print medium is allowed, the system disables a link to that option that goes beyond that limitation and the feature that goes beyond that limitation is not only disabled, but the markup language associated with the feature that goes beyond the printer's limits is not supported by the printing device; col. 8, ln 52-62).

Re claim 3: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

However, Yeung '798 fails to specifically teach a method, wherein the printing device prohibits transmission to the requesting device of the markup language code that is supported by the printing device.

However, this is well known in the art as evidenced by Brossman '921. Brossman '921 discloses wherein the printing device prohibits transmission to the requesting device of the markup language code that is supported by the printing device (i.e. in the system, when the user enters in certain print options in a ticket, the invention only transmits to the user's computer printers that support the specified ticket settings, while prohibiting the transmission of the markup language associated with printers that have markup language that is unsupported by the printing devices; see paragraphs [0027]-[0036]).

Therefore, in view of Brossman '921, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein the printing device prohibits transmission to the requesting device of the markup language code that is supported by the printing device, incorporated in the device of Yeung '798, in order to see if the printer capabilities are available (as stated in Brossman '921 paragraph [0036]).

Re claim 4: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a method, wherein the run-time determination occurs when the printing device boots up or when the request is made for the configuration attributes (i.e. the request or query for the printing device's attributes occurs when a printer driver on the computer (40) accesses a printer-specific data structure on an external printer and compares this data structure to the universal printer data structure definition, which is stored on the requesting or querying computer. The printer-specific data structure or universal data structure, illustrated in figure 3, is a plurality of predetermined data elements used for storing various capabilities supported by one of a plurality of printers. The system performs a determination of what capabilities the printer has and this is sent to the printer driver to compare the printer driver's universal printer data structure definition file and universal printer description file to the printer's associated files; see figs. 1-4 and 6; col. 5, lines 7-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

Re claim 6: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a method, wherein the step of identifying markup language code further comprises the step of identifying markup language code associated with an individual configuration attribute supported by the printing device (i.e. in the system, for every function or attribute that is performed by the printer, a markup language code is associated with the function or attribute. This is illustrated in figures 3 and 4. The printer driver identifies these functions in the system when the printer driver is trying to obtain the correct printer capabilities to communicate correctly to the printer with the printer-specific data structure. The data structure is comprised of XML, which is a markup language; see figs. 3 and 4; col. 5, lines 60-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

Re claim 9: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a method, further comprising the step of generating a device configuration interface to display the printing device's configuration attributes by including markup language code that is associated with the configuration attributes supported by the printing device (i.e. the printing device's attributes are displayed on the user interface for the user to choose what desired settings the user would like to take place on a document. These settings are accompanied by the markup language that are transmitted to the printer driver, so that the printer driver can ensure correct communication with the printer using the same printer-specific data structure described in XML, but display in a format for the user to read and understand. The data of the

UPDF is exchanged between the EEPROM of the printer to the memory of the computing equipment in order to initialize the printer driver on the computer (see col. 10, lines 27-48). The UPDF file is utilized to enable the printer driver to provide an interface to the printer according to the capabilities, characteristics and features of the printer.; see col. 5, lines 2-67, col. 10, lines 27-67, col. 11, lines 1-24 and col. 12, lines 1-24).

Re claim 10: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a method, wherein the step of receiving a request for the printing device's configuration attributes further comprises the step of receiving a request for configuration attributes from a device driver for a printing device (i.e. the request or query for the printing device's attributes occurs when a printer driver on the computer (40) accesses a printer-specific data structure on an external printer and compares this data structure to the universal printer data structure definition, which is stored on the requesting or querying computer. The printer-specific data structure or universal data structure, illustrated in figure 3, is a plurality of predetermined data elements used for storing various capabilities supported by one of a plurality of printers; see figs. 1-4 and 6; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

Re claim 11: Yeung '798 discloses a data structure for printer description file, comprising:

markup language code stored on the printing device (i.e. the markup language code is stored in the ROM (122) or EEPROM (132), in regards to the data elements that represent the capabilities of the printer. The markup language is structured so that the attributes in the system are associated with certain features; see col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67), the markup language code being configured to describe and update the printing device's configuration attributes (i.e. the markup language is used to describe the different functions and attributes of the printer. The XML used is structured in an arrangement that correlates certain features of the printer with XML code. When the determination is made whether the printer-specific data structure matches the universal printer data structure definition, the system checks to see if there are any additional features not accounted for by the universal printer data structure definition (UPDSD), so that these elements may be added to the UPDSD. This is considered as updating the data structure in order to create a better printer-specific data structure in the future; see fig. 2 and 6; col. 3, lines 9-41; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24);

wherein the application is configured to make a run-time determination of which markup language code corresponds to supported configuration attributes of the printing device (i.e. in the device of Yeung, the printing device stores its capabilities in the EEPROM or ROM. The application on the computer is able to access this information and determine, based on the XML data regarding the functions, what features the printer contains; see fig. 2 and 6; col. 3, lines 9-41; col. 5, lines 7-67) and which markup language code corresponds to unsupported configurations attributes of the printing device (i.e. in the system, the UI constraints are identified regarding the printing device and these printer features that are not supported in the printer, but are identified by a recognizing these prevented features of XML. As seen in columns 7 and 8, there are features that may or may not be supported by the printing device. If the data element designates that the feature is not supported, this is an example of code embedded in the printing device as unsupported XML being identified; see fig. 2 and 6; col. 3, lines 9-41; col. 5, lines 42-67 and col. 8, ln 1-62); and

a communication module associated with the printing device (i.e. the communication line (106) is considered as the communication module; see figs. 1 and 2; col. 10, lines 27-67), and the communication module is configured to receive requests for configuration attributes and transmit the markup language code that corresponds to the supported configuration attributes of the printing device (i.e. when the computer (40) tries to access the printer (40) by the communication line (106), it queries the printer's EEPROM or ROM in order to request from or query the printer's memory to compare the printer-specific data structure to the universal printer data structure definition. This example is analogous to the computer asking to see the universal printer data structure definition to compare it to the printer-specific data structure to see if it is valid. Also, during the initialization of the printer driver, the computer may access the memory of the printer to obtain the UPDF and the UPDF is transmitted to the computer. Since the UPDF is in the XML schema, the markup language associated with the configurations are understood to be transmitted to the computer; see fig. 6; col. 3, lines 9-41; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

However, Yeung '798 fails to teach the feature of an embedded application in communication with the printing device and integrated into the printing device, wherein the embedded application is configured to make a run-time determination of which markup language code corresponds to supported configuration attributes of the printing device.

However, this is well known in the art as evidenced by Brossman '921. Brossman '921 discloses the feature of an embedded application in communication with the printing device and integrated into the printing device (i.e. the information handling system can be considered as the embedded application in communication with the printing device since it checks information received from the outside with the capability information on the inside of the printer. The system can also be incorporated within the printer; see paragraph [0036]),

wherein the embedded application is configured to make a run-time determination of which markup language code corresponds to supported configuration attributes of the printing device (i.e. in the use of the information handling system that can be incorporated in the printer, the capabilities of the printer are compared to the selected option by the user. The information handling system makes the determination of the attributes supported by the printer and performs the function of notifying a user the incapability of performing the function or takes the print job and performs the requested function. The function of the printer can happen at run-time since this function occurs when the printer is not in a time of designing the markup language associated with the printer capabilities. Also, the printer capabilities are represented by the XML format in an XML file; see figs. 1, 2 and 4; paragraphs [0015]-[0020], [0023] and [0027]-[0037]).

Therefore, in view of Brossman ‘921, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of an embedded application in communication with the printing device and integrated into the printing device, wherein the embedded application is configured to make a run-time determination of which markup language code corresponds to supported configuration attributes of the printing device in order to see if the printer capabilities are available (as stated in Brossman ‘921 paragraph [0036]).

However, the combination of Yeung ‘798 and Brossman ‘921 fails to teach the feature of wherein the markup language code can enable an active user interface.

However, this is well known in the art as evidenced by Tanimoto ‘219. Tanimoto ‘219 discloses the feature of wherein the markup language code can enable an active user interface (i.e. in the system, the printer is used to send device-setting form data in the HTML or XML form to the requesting device (2), or the client terminal. The client terminal uses the device-setting form data to display a window shown in figure 5 to the user. This is an example of the device-setting form data in HTML or XML being compiled and used to create an active user interface for the user to interact with; see figs. 1-5; paragraphs [0032]-[0036]).

Therefore, in view of Tanimoto ‘219, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein the markup language code can enable an active user interface incorporated in the device of Yeung ‘798, in combination with the features of Brossman ‘921, in order to have the client terminal decode the device-setting form data and show a general browser display to the user (as stated in Tanimoto ‘219 paragraph [0034]).

Re claim 12: The teachings of Yeung '798 in view of Brossman ‘921 and Tanimoto ‘219 are disclosed above.

Yeung '798 discloses a system, wherein the markup language code that corresponds to unsupported configurations attributes is excluded (i.e. in the system, the Yeung reference discloses preventing a user from being able to access options that may exceed the limit of the UI constraints; see col. 8, ln 52-62).

However, Yeung '798 fails to teach the markup language code that corresponds to unsupported configurations attributes is excluded from being transmitted to a device requesting the configuration attributes.

However, this is well known in the art as evidenced by Brossman ‘921. Brossman ‘921 discloses the markup language code that corresponds to unsupported configurations attributes is excluded from being transmitted to a device requesting the configuration attributes (i.e. in the system, when the user enters in certain print options in a ticket, the invention only transmits to the user's computer printers that support the specified ticket settings, while prohibiting the transmission of the markup language associated with printers that have markup language configuration features that are unsupported by the printing devices; see paragraphs [0027]-[0036]).

Therefore, in view of Brossman '921, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of the markup language code that corresponds to unsupported configurations attributes is excluded from being transmitted to a device requesting the configuration attributes, incorporated in the device of Yeung '798, in order to see if the printer capabilities are available (as stated in Brossman '921 paragraph [0036]).

Re claim 13: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a system, wherein the run-time determination of the markup language code refers to a time when the markup language code is executed for the first time (i.e. in the system of Yeung, the computer can access the UPDSD or UPDF in order to verify that the printer driver contains all of the printer's features. When the printer contains any new or additional features, the computer can access these new features on the printer and compare this file with new features to the printer driver's file. Since this will be the first time that these new or additional features will be executed, the printer driver can make sure that proper communication occurs with the printer and its various configurations can be utilized by initializing the printer driver's UPDSD or UPDF files; see col. 3, ln 1-49 and col. 5, ln 7-67).

Re claim 15: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 discloses a system, wherein the markup language code includes XML code (i.e. the markup language in this invention is XML; see appendix A on page 26; col. 3, lines 9-41; col. 5, lines 42-67; col. 6, lines 1-17).

Re claim 16: Yeung '798 discloses a data structure for printer description file, comprising:

a printing means for printing (i.e. the printer (40) in the system has a printer engine (131) to cause an output from the printer; see col. 5, lines 35-41);

a markup language code means for describing configuration attributes (i.e. the universal print data structure file (140) is used to describe the configuration attributes or the printer. This is utilized by the printer driver to configure itself to be able to print on the printer using the correct attribute options; see fig. 2-4 and 6; col. 5, lines 42-67; col. 6, lines 1-25; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24), wherein the markup language code means is stored on the printing means (i.e. the ROM (122) or EEPROM (132) stores the universal printer description file (140) and the universal printer data structure definition file (150) on the printer (40); see col. 5, lines 42-67; col. 6, lines 1-25);

wherein the application means is for making a run-time determination of which markup language code corresponds to the configuration attributes supported by the printing means (i.e. in the device of Yeung, the printing device stores its capabilities in the EEPROM or ROM. The application on the computer is able to access this information and determine, based on the XML data regarding the functions, what features the printer contains; see fig. 2 and 6; col. 3, lines 9-41; col. 5, lines 7-67) and which markup language code corresponds to unsupported configurations attributes of the printing means (i.e. when determining what features are contained on the printing device, the computer also reads the UI constraints which contains in the XML code the unsupported configuration attributes that are prevented from being accessed by the printer driver. In the system, the UI constraints are identified regarding the printing device and these printer features that are not supported in the printer, but are identified by a recognizing these prevented features of XML. As seen in columns 7 and 8, there are features that may or may not be supported by the printing device. If the data element designates that the feature is not supported, this is an example of code embedded in the printing device as unsupported XML being identified; see fig. 2 and 6; col. 3, lines 9-41; col. 5, lines 42-67 and col. 8, ln 1-62); and

a communication module means in the printing means (i.e. the communication line (106) is considered as the communication module; see figs. 1 and 2; col. 10, lines 27-67), wherein the communication port means is for receiving requests for the configuration attributes and transmits configuration attributes supported by the device (i.e. when the computer (40) tries to access the printer (40) by the communication line (106), it queries the printer's EEPROM or ROM in order to request from or query the printer's memory to compare the printer-specific data structure to the universal printer data structure definition. This example is analogous to the computer asking to see the universal printer data structure definition to compare it to the printer-specific data structure to see if it is valid. Also, during the initialization of the printer driver, the computer may access the memory of the printer to obtain the UPDF and the UPDF is transmitted to the computer; see fig. 6; col. 3, lines 9-41; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

However, Yeung '798 fails to teach an embedded application means stored in the printing means, wherein the embedded application means is for making a run-time determination of which markup language code corresponds to the configuration attributes supported by the printing means.

However, this is well known in the art as evidenced by Brossman '921. Brossman '921 discloses the feature of an embedded application means stored in the printing means (i.e. the information handling system can be considered as the embedded application in communication with the printing device since it checks information received from the outside with the capability information on the inside of the printer. The system can also be incorporated within the printer; see paragraph [0036]),

wherein the embedded application means is for making a run-time determination of which markup language code corresponds to the configuration attributes supported by the printing means (i.e. in the use of the information handling system that can be incorporated in the printer, the capabilities of the printer are compared to the selected option by the user. The information handling system makes the determination of the attributes supported by the printer and performs the function of notifying a user the incapability of performing the function or takes the print job and performs the requested function. The function of the printer can happen at run-time since this function occurs when the printer is not in a time of designing the markup language associated with the printer capabilities. Also, the printer capabilities are represented by the XML format in an XML file; see figs. 1, 2 and 4; paragraphs [0015]-[0020], [0023] and [0027]-[0037]).

Therefore, in view of Brossman '921, it would have been obvious to one of ordinary skill at the time the invention was made to have the features of an embedded application means stored in the printing means, wherein the embedded application means is for making a run-time determination of which markup language code corresponds to the configuration attributes supported by the printing means in order to see if the printer capabilities are available (as stated in Brossman '921 paragraph [0036]).

However, the combination of Yeung '798 and Brossman '921 fails to teach the feature of can enable an active user interface.

However, this is well known in the art as evidenced by Tanimoto '219. Tanimoto '219 discloses the feature of can enable an active user interface (i.e. in the system, the printer is used to send device-setting form data in the HTML or XML form to the requesting device (2), or the client terminal. The client terminal uses the device-setting form data to display a window shown in figure 5 to the user. This is an example of the device-setting form data in HTML or XML being compiled and used to create an active user interface for the user to interact with; see figs. 1-5; paragraphs [0032]-[0036]).

Therefore, in view of Tanimoto '219, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of can enable an active user interface incorporated in the device of Yeung ‘798, in combination with the features of Brossman ‘921, in order to have the client terminal decode the device-setting form data and show a general browser display to the user (as stated in Tanimoto ‘219 paragraph [0034]).

Re claim 18: Yeung '798 discloses a data structure for printer description file, comprising:

a computer usable medium having computer readable program code embodied therein for dynamically controlling access to configuration attributes for a printing device (i.e. the EEPROM (132) has reprogrammable memory that stores information that my be provided to the computing equipment (40) to inform the computer of the operational parameters of the printer (40); see col. 5, lines 23-59), the computer readable program code means in the article of manufacture comprising:

computer readable program code for receiving a request for the printing device's configuration attributes (i.e. the request or query for the printing device’s attributes occurs when a printer driver on the computer (40) accesses a printer-specific data structure on an external printer and compares this data structure to the universal printer data structure definition, which is stored on the requesting or querying computer. The printer-specific data structure or universal data structure, illustrated in figure 3, is a plurality of predetermined data elements used for storing various capabilities supported by one of a plurality of printers; see figs. 1-4 and 6; col. 5, lines 42-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24);

computer readable program code for identifying markup language code associated with the configuration attributes supported by the printing device (i.e. in the system, the printing attributes are automatically mapped to an XML structure that arranges the printing attributes in a hierarchal order. When using the example of figure 6 to determine if a printer-specific data structure is valid, the attributes are compared to the attributes in the universal printer data structure definition. The comparison involves the attributes and the markup language associated with the attributes; see figs. 3-6; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24) and markup language code embedded in the printing device unsupported by the printing device (i.e. in the system, the UI constraints are identified regarding the printing device and these printer features that are not supported in the printer, but are identified by a recognizing these prevented features of XML. In the system, the UI constraints are identified regarding the printing device and these printer features that are not supported in the printer, but are identified by a recognizing these prevented features of XML. As seen in columns 7 and 8, there are features that may or may not be supported by the printing device. If the data element designates that the feature is not supported, this is an example of code embedded in the printing device as unsupported XML being identified; see col. 8, ln 1-62); and

computer readable program code for transmitting the markup language code that is associated with the configuration attributes supported by the printing device to the requesting device (i.e. when the computer (40) used in the system accesses the printer-specific data structure through a communication line (106) to the printer (50), after it is discovered that the printer-specific data structure is valid, the data structure is sent or transmitted to the computer (40) for the printer driver to correctly communicate with the printer (50) using the printer-specific data structure. Also, during the initialization of the printer driver, the computer may access the memory of the printer to obtain the UPDF and the UPDF is transmitted to the computer; see figs. 1-3 and 6; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24) and excluding the markup language code that is unsupported by the printing device (i.e. in the system, the UI constraints are used to prevent the user from selecting capabilities or functions that are not supported by the printer. With the prevention of selecting these options, these options are excluded from the user in the system. in the system, the UI constraints are used to prevent the user from selecting capabilities or functions that are not supported by the printer. With the prevention of selecting these options, these options are excluded from the user in the system. The unsupported functions can be the TonerSaves function that may be designated as not supported by the printing device; see col. 8, ln 1-62).

However, Yeung '798 fails to teach the feature of computer readable program code to operate on the printing device for making a run-time determination of configuration attributes supported by the printing device.

However, this is well known in the art as evidenced by Brossman. Brossman discloses the feature of computer readable program code to operate on the printing device for making a run-time determination of configuration attributes supported by the printing device (i.e. in the use of the information handling system that can be incorporated in the printer, the capabilities of the printer are compared to the selected option by the user. The information handling system makes the determination of the attributes supported by the printer and performs the function of notifying a user the incapability of performing the function or takes the print job and performs the requested function. The function of the printer can happen at run-time since this function occurs when the printer is not in a time of designing the markup language associated with the printer capabilities. Also, the printer capabilities are represented by the XML format in an XML file. It is understood that the feature of the information handling system is utilized through program code that can be operated on the printer in the system since most software functions in a printer are operated in program code executed by the CPU of the printer device; see figs. 1, 2 and 4; paragraphs [0015]-[0020], [0023] and [0027]-[0037]).

Therefore, in view of Brossman '921, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of computer readable program code to operate on the printing device for making a run-time determination of configuration attributes supported by the printing device in order to see if the printer capabilities are available (as stated in Brossman '921 paragraph [0036]).

However, the combination of Yeung '798 and Brossman '921 fails to teach the feature of wherein the markup language code can enable an active user interface.

However, this is well known in the art as evidenced by Tanimoto '219. Tanimoto '219 discloses the feature of wherein the markup language code can enable an active user interface (i.e. in the system, the printer is used to send device-setting form data in the HTML or XML form to the requesting device (2), or the client terminal. The client terminal uses the device-setting form data to display a window shown in figure 5 to the user. This is an example of the device-setting form data in HTML or XML being compiled and used to create an active user interface for the user to interact with; see figs. 1-5; paragraphs [0032]-[0036]).

Therefore, in view of Tanimoto ‘219, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of wherein the markup language code can enable an active user interface incorporated in the device of Yeung ‘798, in combination with the features of Brossman ‘921, in order to have the client terminal decode the device-setting form data and show a general browser display to the user (as stated in Tanimoto ‘219 paragraph [0034]).

5. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over Yeung '798, as modified by Brossman ‘921 and Tanimoto ‘219, as applied to claim 1 above, and further in view of Hammond ‘067 (USP 6820067) and Garcia ‘470 (US Pub No 2003/0048470).

Re claim 5: The teachings of Yeung '798 in view of Brossman ‘921 and Tanimoto ‘219 are disclosed above.

Yeung '798 teaches a method, further comprising the steps of parsing an XML tree containing the printing device's configuration attributes (i.e. the DTD file created using the universal printer data structure definition forms a tree-like structure illustrated in figures 3 and 4. This structure is analyzed, or parsed, to find corresponding printing attributes for the printer-specific data structure used to configure the printer driver in the computer (40); see figs. 1-4 and 6; col. 5, lines 60-67; col. 6, lines 1-17; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24) and using the XML tree to display the printing device's configuration attributes (i.e. the printing device's attributes are displayed on the user interface for the user to choose what desired settings the user would like to take place on a document. Since the hierarchal structure of the XML code is used in a UPDF and the UPDF is utilized to initialize the printer driver to create an interface for the printer and the printer's capabilities, the feature of an XML tree used to display the printing device's configuration attributes are performed; see col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

However, Yeung '798 fails to teach using the XML tree to create an HTML page that displays the printing device's configuration attributes.

However, this is well known in the art as evidenced by Hammond '067. Hammond '067 discloses using the XML tree to create an HTML page (i.e. like the reference of Yeung and Brossman, the reference of Hammond processes markup language information (same field of endeavor). However, the reference discloses that an XML file generated by a compiler (28) is read and produced into a set of HTML web pages; see col. 4, lines 18-32).

Therefore, in view of Hammond '067, it would have been obvious to one of ordinary skill at the time the invention was made to create an HTML page incorporated in the device of Yeung '798, as combined with the features of Brossman '921 and Tanimoto '219, in order to have HTML web pages produced from XML documents (as stated in Hammond '067 col. 4, lines 18-32).

However, the combination of Yeung '798, Brossman '921, Tanimoto '219 and Hammond '067 fails to teach the feature of an HTML page that displays the printing device's configuration attributes.

However, this is well known in the art as evidenced by Garcia '470. Garcia '470 discloses the feature of an HTML page that displays the printing device's configuration attributes (i.e. the reference of Garcia is used to perform the feature of communicating a printer's capabilities through markup language, which is similar to the reference of Yeung and Brossman (same field of endeavor). However, in the system of Garcia, the printer web page (202) is page that displays the printer settings (222), toner function (224) and status (230) of the printer. The web page performs the function of displaying the printing device's configuration attributes; see paragraphs [0016], [0021] and [0031]-[0038]).

Therefore, in view of Garcia '470, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of an HTML page that displays the printing device's configuration attributes incorporated in the device of Yeung '798, as combined with the features of Brossman '921, Tanimoto '219 and Hammond '067, in order to have a web page provide access to features of a printer (as stated in Garcia '470 paragraph [0033]).

6. Claims 7, 8 and 17 rejected under 35 U.S.C. 103(a) as being unpatentable over Yeung '798, as modified by the features of Brossman '921 and Tanimoto '219, as applied to claims 1, 11 and 16 above, and further in view of Garcia '470.

Re claim 7: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 teaches a method, wherein the step of receiving a request for the printing device's configuration attributes further comprises the step of receiving the request for the printing device's configuration attributes from a network browser into a printing device over a network (i.e. with the universal print data structure definition file or the universal print describing file being accessed over the internet or LAN, while the user may select desired printing options through a display on the computer 40), this all is analogous to receiving a request for the printing device's attributes from a network browser into a printing device over a network; see fig. 6; col. 10, lines 27-67; col. 11, lines 1-24 and col. 12, lines 1-24).

However, Yeung '798 fails to teach receiving requests from a network browser into printing device's embedded web server.

However, this is well known in the art as evidenced by Garcia '470. Garcia '470 discloses receiving requests from a network browser into printing device's embedded web server (i.e. the printer web page is accessible from a computer workstation (106) through a browser over network (108). The browser is connected to the web page of the printing device's embedded web server (120), which produces the web site; see paragraphs [0021]-[0026] and [0030]-[0038]).

Therefore, in view of Garcia '470, it would have been obvious to one of ordinary skill at the time the invention was made have the feature receiving requests from a network browser into printing device's embedded web server in order to provide access to the features of the printer (as stated in Garcia paragraph [0033]).

Re claim 8: The teachings of Yeung '798 in view of Brossman ‘921 and Tanimoto ‘219 are disclosed above.

Yeung '798 discloses a method, further comprising the step of using a local area network or the World Wide Web of the Internet as the network (i.e. accessing the printer (40) can be performed through an internet connection or over a local or wide are network; see col. 11, lines 1 and 2).

Re claim 17: The teachings of Yeung '798 in view of Brossman ‘921 and Tanimoto ‘219 are disclosed above.

However, Yeung '798 fails to teach a system, wherein the communication module means is an embedded web server.

However, this is well known in the art as evidenced by Garcia ‘470. Garcia ‘470 discloses a system, wherein the communication module means is an embedded web server (i.e. in the system, web server is embedded in the printing device, which communicates with other devices on the network through a hosted web page; see paragraphs [0021]-[0026] and [0030]-[0038]).

Therefore, in view of Garcia '470, it would have been obvious to one of ordinary skill at the time the invention was made to have wherein the communication module means is an embedded web server in order to provide access to the features of the printer (as stated in Garcia paragraph [0033]).

7. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Yeung '798, as modified by the features of Brossman '921 and Tanimoto '219, as applied to claim 11 above, and further in view of Ohara '367 (US Pub no 2003/0182367).

Re claim 14: The teachings of Yeung '798 in view of Brossman '921 and Tanimoto '219 are disclosed above.

Yeung '798 teaches a system, wherein the markup language code includes Meta commands to instruct on including or excluding markup language code at run-time (i.e. in the system, using the UI constraints, the system tells the printer driver what markup language to include in correspondence to a certain feature of the printer for a user to choose from; see col. 8, ln 52-62).

However, Yeung '921 fails to teach Meta commands to a web server.

However, this is well known in the art as evidenced by Ohara '367. Ohara '367 discloses Meta commands to a web server (i.e. the system of Ohara, like the previously applied references above, is used to communicate setting information to a client computer (same field of endeavor). In Ohara, the printer contains a web server and the web server may receive Meta commands on the web page in HTML on the database (14) that instructs the client computer to show the settings of the printing device. The printing device information is shown to the computer when the user requests this information. The printer makes a determination of its capabilities and sends this information to the client computer; see figs. 1-4; paragraphs [0049]-[0058]).

Therefore, in view of Ohara '367, it would have been obvious to one of ordinary skill at the time the invention was made to have the feature of Meta commands to a web server, incorporated in the device of Yeung '798, as modified by the features of Brossman '921 and Tanimoto '219, in order to have the client computer access a printer's web server for the purpose of obtaining information about the printer (as stated in Ohara '367 paragraph [0005]).

(10) Response to Argument

Appellant's arguments filed 5/26/2009 have been fully considered but they are not persuasive. In regards to the claim feature "*wherein the markup language code can enable an active user interface*", the Examiner still believes that this phrase renders the claim indefinite. In MPEP 2173.05 (e), different scenarios of indefiniteness are listed. One scenario states, "*if two different levers are recited earlier in the claim, the recitation of "said lever" in the same or subsequent claim would be unclear where it is uncertain which of the two levers was intended*". This scenario is equivalent to the Appellant's claim language. Earlier in the independent claims, supported and unsupported markup language code associated with configuration attributes of a printer device is recited. Later on, the Appellant's claims recite that "*the markup language code*" enables an active user interface. Without specifying which markup language code enables the active user interface and where the active user interface is being enabled (i.e. at the printer or requesting device), the Examiner would have to read the specification into the claims in order to come to the conclusion of what the Appellant "intended" to claim. As stated in the background of Appellant's invention, printer devices can contain an LCD or interface panel[1]. What in the claim language clearly defines that the active user interface is associated with the requesting device and not the printing device and what markup language code activates this user interface? Therefore, with the above reasoning, the Examiner maintains the above rejection.

Regarding the second argument of the 112 2nd paragraph rejection applied to claims 1 and 18 stating that the phrase "*excluding the markup language code that is unsupported by the printing device*" are indefinite, the Examiner has withdrawn this part of the rejection. In considering the Appellant's arguments, the Examiner has withdrawn this argument and rejection.

When observing the Appellant's remarks related to the 103 rejections, the Examiner noted several assertions. The Appellant asserted that the combined references of Yeung, Brossman and Tanimoto did not disclose the features of (1) identifying both markup language code (hereinafter referred to as MLC) associated with the configuration attributes supported by the printing device and MLC embedded in the printing device unsupported by the printing device, (2) transmitting MLC supported by the printing device <u>and</u> excluding MLC unsupported by the printing device, (3) disabling

[1] See Appellant's spec on pg 2, ll. 9-17.

links to the MLC that is unsupported by the printing device and (4) prohibiting transmission to a requesting device of the markup language code that is unsupported by the printing device. The Examiner respectfully disagrees with these assertions.

When the Examiner interpreted the Appellant's claims, the Examiner interpreted the supported configuration attributes of a printing device to mean that these attributes are installed and are used by the printing device. In terms of the applied prior art, the Yeung reference was applied to the claims to clearly identify the XML code that is associated with the functions on a printer and transmit this information to a connected computer. The transmitted information updated the computer's driver as to the options that are available in the printing process. The Appellant points out that Yeung does identify supported MLC on the printing device and transmits this to another device. Moreover, the Appellant raised the issue that the Yeung reference only identifies configuration attributes supported by the printer storing the file (page 19 of brief). However, the Examiner found portions within the Yeung reference that is contrary to the point at issue. In column 7, ll. 48-50, the specification states, "*The Storages data element 187 indicates whether or not the printing unit being described supports storage units*". If the Storages data element indicates that the storage units are not supported, the system then has identified an attribute, which is coded in XML, not supported by the printing device. Another example is the page protection function mentioned in column 8, ll. 35. If the system designates that the Page protection feature is not performed, the system then identifies an unsupported feature in the printing device. There are other passages that also disclose the feature of having data elements designate whether a certain feature is supported[2]. Once again, since the XML code can designate that certain features are not supported and this is identified by the system, this discloses the feature of identifying both features that are and are not supported by the printing system. It is also important to note that the Brossman reference performs the feature of identifying this feature as well. Therefore, the Examiner still believes the above feature (1) is performed.

Regarding the feature of "*excluding the markup language code that is unsupported by the printing device*", the Examiner still believes this is performed by the Yeung reference as well. The Examiner realized that the scope of this claimed element was fairly broad. Since this limitation did not specify that the MLC was excluded from transmission in the independent claims, the Examiner gave these claims the broadest reasonable interpretation. In paragraphs [0024] and [0025] of the Appellant's specification, there were several examples of excluding MLC. Specifically in paragraph [0024], the specification states, "*Excluding markup language code prevents a user from accessing the markup language code, ...*". Appellant's specification also states in paragraph [0025] that "*Several different mechanisms are available for including or excluding markup language code at run time. One method is that the links to the unsupported markup language code can be disabled*". When viewing the specification, the Examiner interprets excluding the MLC that is unsupported by the printing device as simply not allowing the user to access the unsupported code. When observing the Yeung reference, the UIconstraints data element (212) prevents a user from choosing

[2] See Yeung at col. 8, ll. 1-51.

an option that the printing device does not support. For example, if the printer cannot support the TonerSaves function, the UIconstraints prevents the ability to choose this option[3]. Because the claim limitation does not specifically state what the MLC is being excluded from, like the language recited in claim 3, and the specification speaks of several manners to exclude the MLC, the Examiner believes that the features of (2) excluding MLC that is unsupported by the printing device and (3) disabling links to these unsupported features are performed by the Yeung reference.

Regarding the feature of (4) prohibiting the transmission of unsupported MLC of a printing device to a requesting device, the Examiner used the combination of Yeung with Brossman to disclose this feature. In Brossman, the system contains "*Constraints*" that may be supported by the printer, but is not installed for use by the printer[4]. When a user enters in a job ticket and wishes to output a job, the Brossman system simply returns an error if the job ticket requirements do not meet the functions of the printer, or the job is converted for printing. All this occurs while excluding the unsupported printer capabilities in the XML printer capability file from being transmitted with the exception report to the user computer[5]. Because the Brossman system transmits information that does not include the unsupported features of the capability files to the user computer, the Examiner believes the features of claims 3 and 12 pertaining to the above mentioned claim limitation (4) are performed.

With the above explanation, the Examiner believes that the claim limitations are still performed with the applied references.

[3] Id. at col. 8, ll. 52-62.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Chad Dickerson/

Chad Dickerson

Assistant Examiner (2625)

Conferees:

King Poon (SPE 2625)

/King Y. Poon/

Supervisory Patent Examiner, Art Unit 2625

Twyler Haskins (SPE 2625)

/Twyler L. Haskins/

Supervisory Patent Examiner, Art Unit 2625

[4] See Brossman '921 at paragraphs [0017]-[0025] and pages 3 and 4.

[5] Id. at paragraphs [0015] and [0027]-[0036].